Transferring and Compressing Convolutional Neural Networks for Face Representations

Jakob Grundström[1,2(✉)], Jiandan Chen[2], Martin Georg Ljungqvist[2], and Kalle Åström[1]

[1] Centre for Mathematical Sciences, Lund University, Lund, Sweden
pi07jg8@student.lth.se, kalle@maths.lth.se
[2] Axis Communications, Lund, Sweden
{jiandan.chen,martin.ljungqvist}@axis.com

Abstract. In this work we have investigated face verification based on deep representations from Convolutional Neural Networks (CNNs) to find an accurate and compact face descriptor trained only on a restricted amount of face image data. Transfer learning by fine-tuning CNNs pre-trained on large-scale object recognition has been shown to be a suitable approach to counter a limited amount of target domain data. Using model compression we reduced the model complexity without significant loss in accuracy and made the feature extraction more feasible for real-time use and deployment on embedded systems and mobile devices. The compression resulted in a 9-fold reduction in number of parameters and a 5-fold speed-up in the average feature extraction time running on a desktop CPU. With continued training of the compressed model using a Siamese Network setup, it outperformed the larger model.

1 Introduction

In visual recognition it is rapidly becoming a standard practice to use deep representations composed of layer activations extracted from *Convolutional Neural Networks* (CNNs) as object descriptors, see [1,17]. CNNs are frequent top performers on complex image analysis tasks. However, one of the drawbacks of CNNs is that they require vast amounts of data for training in order to perform well. The CNNs used for this purpose are therefore often pre-trained on huge labeled datasets for generic object recognition containing a large set of object categories, from here on we call those CNNs *generic CNNs*.

Generic CNNs, such as [13,19], can be regarded as general-purpose feature extractors producing *generic object descriptors*, descriptors that may also constitute good representations for domains other than the *source domain*.

Even though a generic CNN usually perform well in domains other than those it was trained for, it still lacks specificity. In many cases the object representations can be further improved by adapting the CNN to the *target domain*, as done in [1] and which led to state of the art results on 16 visual recognition benchmarks.

A. Campilho and F. Karray (Eds.): ICIAR 2016, LNCS 9730, pp. 20–29, 2016.
DOI: 10.1007/978-3-319-41501-7_3

The process of transferring a generic CNN to a new data domain is often called *fine-tuning* and is a way to do transfer learning. Fine-tuning involves training a CNN structure initialized with weights from the pre-trained generic CNN and using data from the target domain.

To recognise subjects in images of arbitrary angle, position, lighting and other variables is a complex task which requires large CNNs with many layers for training. To evaluate a trained CNN model on unseen data the entire CNN structure is needed. This is much more time efficient than training. However, a real-time application in an embedded device with limited computational resources is still a challenge for CNNs with many layers. A way to handle this is *model compression* [4,9] where a large CNN can be trained into a smaller model.

A popular application of machine learning is face verification; its purpose is to confirm or deny a claimed identity. In a pair-wise formulation the problem is to decide if two face images of previously unseen identities are of matching or non-matching identities.

In this work we have, firstly, investigated face verification based on deep representations extracted from CNNs to find an accurate and compact face descriptor using a restricted amount of face image training data exclusively from publicly available datasets. Secondly, we have reduced the computational demands of the CNN architectures by the use of model compression in order to produce a faster and less resource-demanding feature extractor.

2 Related Works

Solving challenging *face identification* problems has proven a successful strategy to produce accurate face descriptors [22,24,26]. In [22] multiple CNNs are trained on a face identification problem including 10 000 identities. According to the authors training on such a hard classification problem, with many identities and many examples per identity, is crucial for the success of a descriptor. The recent work of [18] takes large-scale training to its extreme and use a non-public training set including roughly 8 million persons. With only a limited amount of publicly available face image data, we instead investigate how competitive a face descriptor can be when pre-trained on large-scale generic object recognition and then fine-tuned using significantly less data from the target domain.

Our fine-tuning approach builds on a CNN architecture presented in [7] and use the pre-trained weights of the *CaffeNet* CNN [11], an architecture similar to the renowned AlexNet CNN [13]. A similar fine-tuning setup was previously used in [12] to transfer-learn a generic CNN to recognise Flickr image categories.

The central idea of model compression is that the function learned by a large and slow but accurate model can be approximated with a fast and compact model. The compact model contains fewer parameters and ideally allows for faster processing. In [4] this idea was applied to compress large ensembles of machine learning models into compact neural networks. The concept is further investigated in [9] but with focus on transfer learning of the generalization properties from a complex model to a simpler one. For a well-performing model on

one-of-many classification task and given a training example, most class probabilities would be close to zero. To increase the influence of those small values (i.e. the *dark knowledge* that encode relative likelihoods of the incorrect classes) in the cross-entropy loss function the authors smoothed the class probabilities of the complex model and used those as softened targets for learning the simpler model. The results of [2] shows that deep neural networks many times can be compressed into shallow neural networks. The compressed nets matched the performance of deep architectures on both phoneme recognition and image recognition. The shallow models were trained by regressing the logits of the class probabilities with l_2-loss.

With a pair-wise training in a *Siamese Network* setup it is possible to also leverage similarity. Examples of this is training done using the *Contrastive Loss* [6,8,24]. A recent class of face verification algorithms employ both discriminative and similarity-based objectives. For instance [21,23] use a combination of classification and verification supervision and show great results in face verification. Another way to perform supervised learning with both similarity-based and discriminative objectives is described by [3], where the authors present a method for learning a cross-domain similarity by combining Contrastive Loss and cross-entropy loss. We apply this approach to face verification.

3 Method

The investigated approaches include: (1) using a generic CNN as a general-purpose feature extractor; (2) training a CNN from scratch exclusively on face recognition data; (3) fine-tuning a generic CNN with face recognition data; (4) transferring a fine-tuned generic CNN to a compressed CNN architecture with model compression; (5) transferring a fine-tuned generic CNN to a compressed CNN architecture with model compression and train it as a Siamese Network.

In (1) we used the pre-trained generic CNN directly without modification, using the weights of CaffeNet [11], and extracted its 4096-dimensional last hidden layer activations as face descriptors. For (2) a CNN was trained from scratch exclusively on face recognition data and with random initialization of the weights. The learning rates were increased compared to fine-tuning and the network was allowed to train for longer, but stopping on the same minimum learning rate. Here we used the same architecture as the fine-tuned model (3). The approaches (3–5) are described in detail in the following sections.

For all the reported face verification results we used an implementation of the *Joint Bayesian* classifier [5] applied on CNN descriptors projected into a 200-dimensional space through *Principal Component Analysis* (PCA). The dimension 200 was empirically chosen with the aim to create a both compact and accurate descriptor.

Fine-Tuning. The CNN architecture of the fine-tuning setup is based on CaffeNet [11] but with a dimensionality reduction layer to facilitate for a more compact descriptor. In summary we perform the following modifications:

- **Replace output layer** with a new fully connected layer with as many hidden units as identities in the fine-tuning dataset.
- **Adjust learning rates** to learn the new layers faster than the pre-trained layers. Like [12], we set the initial learning rate to 0.001 and decreasing with a step size of 10 000 iterations with a factor of 10.
- **Insert bottleneck layer**. A fully-connected layer with number of hidden units reduced to 1024 was inserted before the output layer.

The inserted layer and the replaced output layer were randomly initialized while the rest of the CNN was initialized with weights pre-trained on large-scale generic object recognition. The face descriptors are then extracted as the activations of the 1024-dimensional last hidden layer from the fine-tuned CNN.

Model Compression. Fine-tuning from pre-trained weights sets some architectural constraints on a CNN, for example layers initialized with pre-trained weights need to conform to the dimensions of the pre-trained model. Initializing the bottom CNN layers with pre-trained weights for instance implicates a certain input size, in our case $256 \times 256 \times 3$ dimensions which is considered large for typical face recognition applications. The fine-tuned CNN model may be overparameterized when transferring to a less complex problem, as we do when fine-tuning for face identification from generic object recognition.

Model compression presents a way to overcome these problems. Especially, we use it to train a compressed model with a more light-weight CNN architecture and with the input size reduced to $64 \times 64 \times 3$, or effectively $56 \times 56 \times 3$ because of the subcropping data augmentation we use. We apply model compression to the fine-tuned CNN (3).

Training. The model compression setup follows the idea of [2] and consists of a complex model (3), acting as a teacher, and a simple model with reduced complexity (4) (see Table 1). During training both models are given identical images as input but scaled to match the input sizes of each CNN. The smaller model is then learned by performing simultaneous regression and classification, as depicted in Fig. 1, while the parameters of the larger model $\tilde{\Theta}$ are freezed by setting the learning rate to zero. The regression uses the class log-probabilities from the complex model $\tilde{z}$ as targets and is formulated using the *Euclidean Loss* as follows

$$\mathcal{L}(\Theta) = \frac{1}{2N} \sum_{n=1}^{N} ||\tilde{z}^{(n)} - z^{(n)}||_2^2, \tag{1}$$

over a mini-batch of N images $\{\mathbf{I}_n : n \in [1, N]\}$ and where z denote the log-probabilities predicted by the compressed model with parameters Θ.

The classification objective for the compressed model is formulated as common practice for CNNs, with cross-entropy loss according to the class label $l^{(n)}$

$$\mathcal{L}(\Theta) = -\frac{1}{N} \sum_{n=1}^{N} \log(p_{l^{(n)}}^{(n)}), \tag{2}$$

where the probability $p_k^{(n)}$ for a class $k \in \{1, K\}$ of an input image $\mathbf{I}_n$ is computed from the class log-probabilities z using softmax as follows

$$p_k^{(n)} = \frac{\exp\left(z_k^{(n)}\right)}{\sum_{j=1}^{K} \exp\left(z_j^{(n)}\right)}. \tag{3}$$

The training was done in a three stage process gradually increasing the weight of the Euclidean Loss (in steps 10^{-9}, 10^{-6} and 10^{-3}) while keeping the classification loss at unit weight.

The architecture of the compressed model is inspired by [20], using consecutive convolutional layers with 3×3 kernels including ReLU-activation functions rather than larger kernels. The convolutional layers use 1 pixel padding such that the dimensions are preserved and the max-pooling use 2×2 kernels with a stride size of 2.

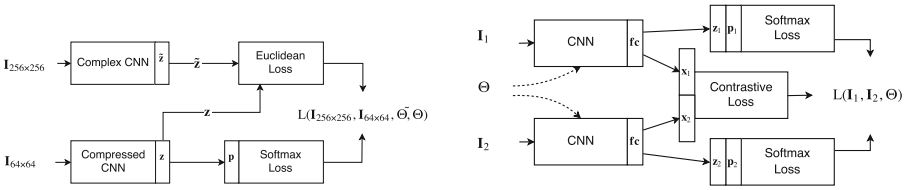

Fig. 1. Setups for model compression *(left)* and Siamese Network training *(right)*

The design (4–5) shown in Table 1 resulted in a compressed model with $6.7M$ parameters (not including bias parameters), which is close to a 9-fold reduction from the renowned AlexNet and the CaffeNet with their $60M$ parameters.

Siamese Network. To further improve the compressed model we turned to Siamese Network techniques (5) and adopted the pair-wise training setup using both identification and similarity illustrated in Fig. 1. A setup, in which we use a combination of softmax loss and contrastive loss, similar to [3]. The idea is to learn an embedding space for faces such that similar faces are pulled closer to each other and faces of different identities pushed away from each other if within a certain constant margin. This can be formulated mathematically with the *Contrastive Loss* function

$$\begin{aligned} \mathcal{L}(\Theta) = & \frac{1}{2N} \sum_{n=1}^{N} y||x_1^{(n)} - x_2^{(n)}||_2^2 + \\ & (1-y)\max(m - ||x_1^{(n)} - x_2^{(n)}||_2, 0)^2, \end{aligned} \tag{4}$$

with face embedding vectors x, a constant margin m and similarity $y \in \{0, 1\}$ [6,8].

Table 1. CNN architecture of the proposed compressed model

CNN layer	Kernel	Channels	Dimension	Parameters
Input (Subcropping)		3	56 × 56	
conv1 (relu)	3 × 3	32	56 × 56	864
conv2 (relu)	3 × 3	32	56 × 56	9216
pool1	2 × 2		28 × 28	
conv3 (relu)	3 × 3	64	28 × 28	18432
conv4 (relu)	3 × 3	64	28 × 28	36864
pool2	2 × 2		14 × 14	
conv5 (relu)	3 × 3	128	14 × 14	73728
conv6 (relu)	3 × 3	128	14 × 14	147456
pool3	2 × 2		7 × 7	
fc (dropout)		1024	1 × 1024	6422528
Total				6709088

The input to the training setup is a face image pair $(\mathbf{I}_1, \mathbf{I}_2)$ of matching or non-matching identities. The images are forward-passed through identical CNNs sharing the same parameters Θ producing embedding vectors (x_1, x_2) and the estimated class probabilities (p_1, p_2). The embedding vectors x are 256-dimensional linear projections of the 1024-dimensional feature layer and can be seen as a second output layer. The softmax loss is formulated according to the predicted class probabilities of the face identity output layer and labels of each pair-member respectively, while the contrastive loss is formulated using the embeddings (x_1, x_2) and the known binary similarity y.

Data. The training data used for CNN fine-tuning, model compression and for learning the Joint Bayesian is a combination of *FaceScrub* [15] and *MSRA-CFW* [25][1]. To avoid overlapping subjects w.r.t. the chosen evaluation dataset LFW [10] we applied blacklisting based on subject names. The dataset was expanded by data augmentation: adding noise, color augmentation and foveation. We balanced the subjects to have a minimum of 200 augmented samples per subject. The final dataset is based on 72 106 original images and were increased to 526 602 by data augmentation.

4 Results

The five presented approaches were all evaluated on the LFW benchmark [10] according to the *Unrestricted Labeled Outside Data* protocol. Figure 2 shows

[1] FaceScrub and MSRA-CFW were downloaded from individual URLs and many images failed to download or were corrupt. For MSRA-CFW we applied a haar-cascade face detector on the downloaded images and created weak annotations.

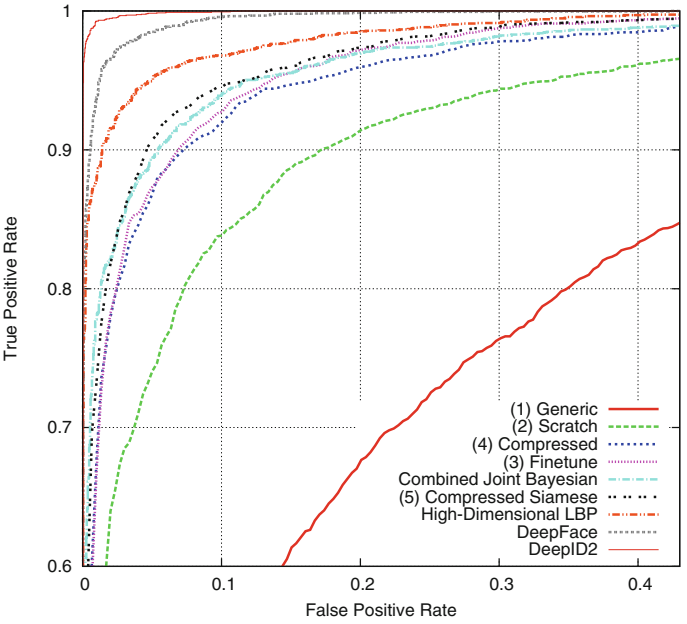

Fig. 2. Receiver-Operating-Characteristics (ROC) curves for the LFW face verification benchmark under the *Unrestricted Labeled Outside Data* protocol

the ROC curves of our approaches and also includes some of the referenced algorithms for comparison. In Table 2 we report the LFW scores as verification accuracy and standard deviation.

Using the generic CNN (1) as general purpose feature extractor gave worse results than training a CNN exclusively on a face image dataset of 777 subjects (2) (Sect. 3), the resulting LFW verification accuracies were 73.5 % and 86.8 % respectively. Fine-tuning a pre-trained CNN (3), transfer learning to the face domain, improved the results significantly and gave an accuracy of 91.6 %. Notably the results from (3) was transferred using model compression to a smaller, more efficient architecture (4) without significant loss in accuracy, 91.4 %.

With (5) the accuracy of the compressed model was improved by continued training using a Siamese Network setup with Contrastive Loss such that it outperformed the large fine-tuned CNN (3). This resulted in our best performing architecture with 92.9 % verification accuracy.

We performed a timing experiment extracting features from 1000 images with both the fine-tuned (3) and the compressed model architecture (4–5). The results were 191.46 ms and 37.72 ms respectively. This constitutes a 5.08-fold speed-up performing feature extraction using the compressed model.

Table 2. Verification accuracies and standard deviations of our five approaches on the LFW benchmark

CNN architecture	Verification accuracy (%)	Std.Dev. (%)
(1) Generic	73.5	1.45
(2) Scratch	86.8	1.59
(3) Fine-tune	91.6	1.24
(4) Compressed	91.4	1.68
(5) Compressed Siamese	92.9	1.42

5 Discussion

The results presented in Sect. 4 shows that good face verification accuracy can be achieved using a comparably small amount of labeled face data, 72 106 images, and with a reasonably compact face descriptor of 200 dimensions.

Our best performing descriptor (5) achieve verification accuracy on the LFW benchmark comparable to the *Combined Joint Bayesian* [5] (see Fig. 2), which use a similar amount of data, 99 773 images, and combines the score of 4 Joint Bayesian classifiers. Our approach uses only a single model and a more compact descriptor. Despite the successful applications of CNN-based transfer learning approaches in many visual tasks [1] we see that our results do not quite match the current state-of-the-art algorithms in face verification, which give accuracies up to 99.79 % on the LFW benchmark. Even so, it should be noted that most of the state-of-the-art algorithms use target domain data several magnitudes larger than the training set we used in this work.

Fine-tuning from a generic CNN provides a way to produce accurate descriptors also in the face recognition domain, even though the domain of face images intuitively is very distant from the domain of generic objects. Comparing the results of (2) and (3) we see that when limited amount of data is available in the target domain fine-tuning a generic CNN can improve generalization compared to training from scratch exclusively on data from the target domain.

Moreover, the benefits of the fine-tuning is transferred also to the compressed model. With model compression (4) the computational demands of the feature extraction could be significantly reduced with just a minor loss in accuracy.

Encouraging results from two very recent papers show that: state-of-the-art face verification can be achieved training a CNN model similar to but slightly deeper than (5) from scratch using more data [16], $2.6M$ images; model compression can be successfully applied also to compress state-of-the-art face verification algorithms [14]. Our results together with these two papers strengthen our hypothesis that fine-tuning combined with model compression makes it possible to achieve accuracy competitive to state-of-the-art in face verification using an efficient, compact model and less data.

6 Conclusion

Transfer learning by fine-tuning from generic CNNs has been shown to be a suitable approach when only a limited amount of data is available in the target domain. Our most accurate approach compares to face verification algorithms trained on a similar amount of data and with a more compact representation. However, the accuracy does not reach that of the state-of-the-art algorithms trained on target domain datasets of a significantly greater magnitude.

By using model compression we reduced the model complexity without significant loss in accuracy and produced a feature extractor more suitable for real-time use. The compression resulted in a 9-fold reduction in number of parameters and a 5-fold speed-up in the average feature extraction time on a quad-core CPU. Additionally, the compressed model gave higher accuracy than the larger model after continued training in a Siamese Network, 92.9 % on the LFW benchmark. This result suggests that model compression is a step towards deployment on platforms with less computational resources, such as embedded systems and mobile devices.

References

1. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: From generic to specific deep representations for visual recognition. CoRR abs/1406.5774 (2014). http://arxiv.org/abs/1406.5774
2. Ba, L.J., Caurana, R.: Do deep nets really need to be deep? CoRR abs/1312.6184 (2013). http://arxiv.org/abs/1312.6184
3. Bell, S., Bala, K.: Learning visual similarity for product design with convolutional neural networks. ACM Trans. Graph. **34**(4), 98:1–98:10 (2015). http://doi.acm.org/10.1145/2766959
4. Bucila, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 Aug 2006, pp. 535–541 (2006)
5. Chen, D., Cao, X., Wang, L., Wen, F., Sun, J.: Bayesian face revisited: a joint formulation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 566–579. Springer, Heidelberg (2012). http://dx.doi.org/10.1007/978-3-642-33712-3_41
6. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 539–546 (2005)
7. Grundström, J.: Face verification and open-set identification for real-time video applications (2015). Student Paper
8. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1735–1742 (2006)
9. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015). http://arxiv.org/abs/1503.02531
10. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report 07–49, University of Massachusetts, Amherst, October 2007

11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding (2014). arXiv preprint arXiv:1408.5093
12. Karayev, S., Hertzmann, A., Winnemoeller, H., Agarwala, A., Darrell, T.: Recognizing image style. CoRR abs/1311.3715 (2013). http://arxiv.org/abs/1311.3715
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates Inc. (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
14. Luo, P., Zhu, Z., Liu, Z., Wang, X., Tang, X.: Face model compression by distilling knowledge from neurons (2016). http://personal.ie.cuhk.edu.hk/~pluo/pdf/aaai16-face-model-compression.pdf
15. Ng, H., Winkler, S.: A data-driven approach to cleaning large face datasets. In: ICIP14, pp. 343–347 (2014)
16. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference (2015)
17. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. CoRR abs/1403.6382 (2014). http://arxiv.org/abs/1403.6382
18. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. CoRR abs/1503.03832 (2015). http://arxiv.org/abs/1503.03832
19. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: integrated recognition, localization and detection using convolutional networks. In: International Conference on Learning Representations (ICLR 2014). CBLS, April 2014. http://openreview.net/document/d332e77d-459a-4af8-b3ed-55ba
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014). http://arxiv.org/abs/1409.1556
21. Sun, Y., Wang, X., Tang, X.: Deep Learning Face Representation by Joint Identification-Verification. Ph.D. thesis, arXiv (2014). http://arxiv.org/abs/1406.4773
22. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: Computer Vision and Pattern Recognition, pp. 1891–1898. IEEE (2014)
23. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. CoRR abs/1412.1265 (2014). http://arxiv.org/abs/1412.1265
24. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
25. Zhang, X., Zhang, L., Wang, X.J., Shum, H.Y.: Finding celebrities in billions of web images. IEEE Trans. Multimedia **14**(4), 995–1007 (2012)
26. Zhou, E., Cao, Z., Yin, Q.: Naive-deep face recognition: touching the limit of LFW benchmark or not? CoRR abs/1501.04690 (2015). http://arxiv.org/abs/1501.04690